



TAMPEREEN
AMMATTIKORKEAKOULU

ESINEIDEN INTERNET JA IOT-ALUSTAT

Jari Uppa

Opinnäytetyö
Syyskuu 2017
Tietojenkäsittely



TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietojenkäsittely

UPPA, JARI:
Esineiden internet ja IoT-alustat

Opinnäytetyö 41 sivua, joista liitteitä 2 sivua
Syyskuu 2017

Tämän opinnäytetyön tarkoituksena oli tutustua esineiden internetiin, sen peruskäsitteisiin sekä selvittää, kuinka esineiden internetissä toimivia järjestelmiä voidaan teknisesti toteuttaa. Lisäksi selvitettiin markkinoilla olevien IoT-alustojen tarjontaa ja tutustuttiin tarkemmin avoimen lähdekoodin ThingsBoard-sovellusalustaan.

Opinnäytetyössä selvitettiin esineiden internetin osa-alueita, niihin liittyviä käsitteitä ja erilaisia yleisesti käytettyjä yhteys- ja sovellusteknologioita. Työssä tutustuttiin myös käytännön esimerkein erilaisiin esineiden internetin sovelluksiin ja käyttötapoihin. Lisäksi tutustuttiin markkinoilla oleviin valmiisiin IoT-alustoihin ja -tuotteisiin, joita voidaan hyödyntää IoT-järjestelmien rakentamisessa.

Työssä perehdyttiin ThingsBoard IoT-sovellusalustaan, jonka tarjoama kokonaisuus sisältää rajapinnat IoT-laitteiden liittämiseen ja hallintaan, toiminnot laitteiden välittämän informaation analysointiin ja visualisointiin sekä rajapinnat järjestelmän ulkopuolisten sovellusten liittämiseen. ThingsBoard asennettiin tutustumista varten Linux-palvelimelle ja siihen liitettiin Java-ohjelmointikielen avulla toteutettuja simuloituja IoT-laitteita. Laitteiden välittämää ja ThingsBoard-järjestelmään tallennettua dataa visualisoitiin järjestelmän tarjoamilla työkaluilla.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Business Information Systems

UPPA, JARI:
Internet of Things and IoT Platforms

Bachelor's thesis 41 pages, appendices 2 pages
September 2017

The purpose of this thesis was to study and document the concepts and technologies related to Internet of Things and find out how IoT systems could be implemented. In addition, IoT platforms readily available on the market were explored and the ThingsBoard platform was selected for closer inspection.

Different use cases and real-life examples were examined in order to make understanding the concepts of Internet of Things easier. The different technical aspects including network and communication protocols used in IoT systems were also studied.

The ThingsBoard IoT platform was examined in practice by installing it on a server running a Linux operating system and connecting it with a few simple simulated IoT devices which were implemented using the Java programming language. The functionalities and tools provided by ThingsBoard were used for creating device and end user configurations and dashboards containing visualizations of data sent by the connected devices.

Key words: internet of things, iot, platform

SISÄLLYS

1	JOHDANTO.....	7
2	ESINEIDEN INTERNET	8
2.1	Mikä on esineiden internet.....	8
2.2	Nykytilanne ja tulevaisuus.....	9
2.3	Esimerkkejä sovelluksista.....	10
3	IOT-JÄRJESTELMÄN ARKKITEHTUURI	12
3.1	Yleinen arkkitehtuuri	12
3.2	Sensorit, laitteet, ja aktuaattorit	14
3.3	Verkkoteknologiat	15
3.4	Sovellustason viestiprotokollat.....	17
4	IOT-ALUSTAT.....	18
4.1	Mikä on IoT-alusta	18
4.2	IoT-alustojen tyypit	19
4.3	IoT-alustan vaatimukset.....	20
5	ESIMERKKEJÄ IOT-ALUSTOISTA	21
5.1	Kaupalliset tuotteet	21
5.1.1	AWS IoT Platform.....	21
5.1.2	Google Cloud IoT Core	22
5.1.3	Arm Mbed IoT Platform	23
5.2	Open Source –tuotteet	23
5.2.1	Kaa IoT Platform	24
5.2.2	BeagleBoard.....	24
6	THINGSBOARD	26
6.1	Esittely	26
6.1.1	Laitevaatimukset.....	26
6.1.2	Arkkitehtuuri.....	27
6.1.3	Käyttäjät ja laitteet	28
6.1.4	Valvontanäkymä ja vimpaimet	29
6.1.5	Järjestelmään liitettyjen laitteiden ohjaaminen.....	29
6.2	Käytännön tutustuminen järjestelmään.....	30
6.2.1	Laitesimulaattori	31
6.2.2	Laitteiden rekisteröinti	31
6.2.3	Valvontanäkymä	32
6.2.4	Huomioita alustasta.....	34
7	POHDINTA	36
	LÄHTEET.....	38

LIITTEET	40
Liite 1. Laitesimulaattori, MQTT-viestin lähetys	40
Liite 2. Laitesimulaattori	41

LYHENTEET JA TERMIT

Aktuaattori	Toimilaite, fyysisen toiminnon suorittava laite
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP over Transport Layer Security
IoT	Internet of Things, esineiden internet
JSON	JavaScript Object Notation
MQTT	Message Queue Telemetry Transport
QoS	Quality of service
REST	Representational state transfer
RPC	Remote Procedure Call
SASL	Simple Authentication and Security Layer
TLS	Transport Layer Security
XMPP	Extensible Messaging and Presence Protocol

1 JOHDANTO

Internet of Things eli esineiden internet on ollut jo jonkin aikaa yksi teknologia-alan puhutuimpia ja suosituimpia aiheita, mutta asian sisäistäminen ja ymmärtäminen ei ole kovinkaan yksinkertainen tehtävä. Varsinkin sukeltaminen pintaa syvemmälle tekniikan tarjoamiin mahdollisuuksiin ja käyttötarkoituksiin, mahdollisiin ongelmiin sekä teknisiin yksityiskohtiin vaatii tarkkaa perehtymistä ja asian selvittämistä. Koska alue kuitenkin on verrattain nuori ja kehittyy kovaa vauhtia, yhteen koottua ja helposti omaksuttavaa tietoa löytyy kuitenkin toistaiseksi vielä suhteellisen vähän. Esineiden internetistä ei ole valmiita piirustuksia ja ainoastaan yhtä oikeaa tapaa toteuttaa se, vaan erilaisia toimijoita on jo tällä hetkellä markkinoilla erittäin runsaasti ja lisää syntyy koko ajan, jokaisen tuodessa markkinoille oman ratkaisunsa ja tuotteensa omista lähtökohdistaan yhteisten standardien puuttuessa.

IoT-maailmaan liittyy monia erilaisia tasoja; uuden teknologian tarjoamat mahdollisuudet ja sen aiheuttamat haasteet ja ongelmat, laitetaso johon kuuluvat varsinaiset ”esineet” ja niiden yhteydenpidon muihin laitteisiin mahdollistavat teknologiat, sekä ohjelmistot ja taustajärjestelmät, joiden avulla kerätyt suuret tietomassat käsitellään loppukäyttäjiä hyödyttävään muotoon.

Opinnäytetyön tavoitteena on tutustua IoT-maailmaan ja esitellä IoT-teknologia ja sen tarjoamat mahdollisuudet yleisellä tasolla. Lisäksi perehdytään IoT-järjestelmän vaatimuksiin, teknologioihin sekä arkkitehtuuriin ja luodaan katsaus tällä hetkellä markkinoilla oleviin IoT-alustoihin- ja järjestelmiin. Tarkoituksena on koota tietopaketti, jonka perusteella lukija saa käsityksen esineiden internetistä ja teknisistä ratkaisuista joiden avulla IoT-järjestelmiä voidaan rakentaa. Lisäksi tutustutaan hieman tarkemmin avoimen lähdekoodin ThingsBoard IoT-sovellusalustaan. Tarkoituksena ei ole perehtyä liian syvällisesti teknisiin yksityiskohtiin, eikä ottaa kantaa minkään tuotteen puolesta vaan antaa lukijalle lähtökohta omatoimiseen asiaan perehtymiseen.

Opinnäytetyön aihe pohjautuu kirjoittajan omaan mielenkiintoon aihetta ja erityisesti sen teknisiin ominaisuuksiin. Valmiiden IoT-alustojen ja -tuotteiden parantunut tarjonta ja saatavuus ovat madaltaneet kynnystä omien kokeilujen aloittamiseen ja tämän opinnäytetyön tekemisen ohessa tapahtuva tutkimustyö auttaa jatkamaan asiaan perehtymistä.

2 ESINEIDEN INTERNET

2.1 Mikä on esineiden internet

Esineiden internetillä tarkoitetaan laitteiden kytkemistä internetverkkoon, jonka välityksellä ne lähettävät joko omasta tilastaan tai ympäristöstään erilaisilla sensoreilla keräämäänsä tietoa tyypillisesti pilvipalveluun, jossa laitteiden välittämää raakadataa voidaan analysoida ja visualisoida. Laitteet voivat kommunikoida verkon välityksellä sekä toisten laitteiden, että ihmisten kanssa. Esineiden internetin liikenne on tyypillisesti kaksisuuntaista ja se mahdollistaa lisäksi laitteiden etäohjauksen ja niiden ohjelmistojen päivittämisen.

Verkkoon liitettävien laitteiden kirjo vaihtelee teollisuuden ja kotitalouksien koneista aina vaateantureihin saakka. On syytä huomata, että esineiden internetiin ajatellaan kuuluvan laitteet, joita ei aikaisemmin ole kytketty verkkoon, joten esimerkiksi tietokoneita ja matkapuhelimia ei yleensä lueta IoT-laitteiksi. Laitteiden keräämä data muodostaa arvokkaan tietovaraston, jota voidaan hyödyntää useilla eri tavoilla. Sen perusteella voidaan säätää laitteiden toimintaa, ennustaa esimerkiksi mahdollista rikkoutumista tai vaikkapa valvoa laitteen kantajan terveydentilaa.

Esineiden internet mahdollistaa etäohjattavien älykkäiden laitteiden yhdistämisen tietokoneisiin ja tietojärjestelmiin aivan uudella tavalla ja tämä aukaisee uusia mahdollisuuksia kehittää ihmisten jokapäiväistä elämää helpottavia sovelluksia. Lisäksi teollisuuden ja yhteiskunnan toimintojen odotetaan kokevan suuria muutoksia lisääntyvän automaation ja tehokkuuden vuoksi. IoT-teknologian kehittyessä nopeasti sen vaikutus näkyy laajasti monilla eri tavoin, eikä sen tarjoamia kaikkia mahdollisuuksia varmasti ole vielä ymmärretty.

Tällä hetkellä vaikuttaa siltä, että uusien sovellusalueiden suhteen vain mielikuvitus on rajana. Asia ei kuitenkaan ole aivan ongelmaton ja esineiden internetin mahdollisina ongelmina ovat nousseet esille esimerkiksi yksityisyyteen ja tietoturvaan kohdistuvat huolet. Huonosti suojattuja IoT-laitteita on käytetty jo nyt osana massiivisia bottiverkkoja, joilla on aiheutettu vakavia häiriöitä internet-verkon toimintaa.

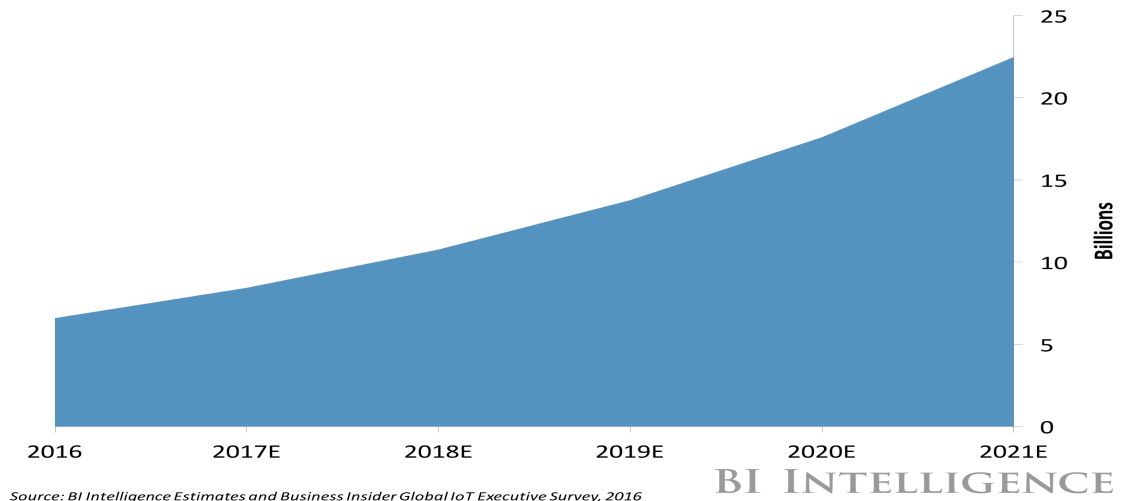
2.2 Nykytilanne ja tulevaisuus

Internet-verkkoon kytkettyjen laitteiden tarkkaa määrää ei kukaan pysty varmuudella arvioimaan. Selvää kuitenkin on, että määrä on jo nyt huomattavan suuri ja se kasvaa lähivuosina edelleen kovaa vauhtia. On esitetty arvioita, että verkkoon liitettyjen älykkäiden laitteiden määrä oli vuonna 2016 noin 6,5 miljardia ja sen odotetaan kasvavan vuoteen 2021 mennessä 22,5 miljardiin laitteeseen (Business Insider 2017). Kuvassa 1 nähdään ennuste laitemäärän kehittymisestä.

Syitä esineiden internetin kovaan kasvuun on monia. Kotitalouksien kohdalla älykotien odotetaan olevan kasvava trendi tulevina vuosina ja älykkäät verkkoon kytketyt laitteet ovat sen olennainen osa. Kodeissa yleistyvät kodin eri toimintoja valvovat ja säättävät järjestelmät, kuten esimerkiksi erilaiset valvonta- ja turvajärjestelmät ja kodin lämmitystä ja valaistusta hoitavat ratkaisut. Yhdysvalloissa tehdyn kyselyn mukaan 71 % vastaajista odottaa jokaisesta kodista löytyvän vähintään yksi älykäs verkkoon kytketty laite vuonna 2025 (Intel 2015). Terveystieteiden sektorilla esineiden internetin markkinoiden arvioidaan myös kasvavan ja arvons olevan vuonna 2020 jo noin 117 miljardia dollaria (Forbes 2015).

Yritysten näkökulmasta edut ovat ilmeiset. Laitteiden keräämän datan analysoinnilla saadun tiedon perusteella on mahdollista alentaa toimintakustannuksia ja lisätä tuottavuutta. Lisäksi esineiden internet mahdollistaa uusien tuoteinnovaatioiden kehittämisen ja sitä kautta uusien markkina-alueiden syntyminen (Business Insider 2016). Yritysten arvioidaan sijoittavan esineiden internetin kehittämiseen ja siihen liittyvien palveluiden hankkimiseen vuoteen 2020 mennessä jopa 250 miljardia euroa. (Boston Consulting Group 2017), joten alueella on näköpiirissä valtavat markkinat sekä laitteiden ja niihin liittyvien ohjelmistojen ja palveluiden osalta.

FORECAST: IoT Device Installation Base
Global, 2016-2021



KUVA 1. Ennuste käytössä olevien IoT-laitteiden määrän kehityksestä (Business Insider 2017)

2.3 Esimerkkejä sovelluksista

Esineiden internet tuntuu mahdollistavan loputtomasti erilaisia käyttötapoja lähes kaikilla kuviteltavissa olevilla alueilla. Sovellukset vaihtelevat yksinkertaisista muutamia yksittäisiä laitteita valvovista järjestelmistä erittäin laajoihin ja monimutkaisiin miljoonia sensoreita kattaviin järjestelmiin, jotka generoivat suuren määrän dataa. Esineiden internetin tuomista mahdollisuuksista pääsee parhaaseen käsitykseen oikeiden esimerkkien avulla, alla muutamia esimerkkejä eri sovellusalueilta.

Lentokonevalmistaja Bombardier on esitellyt C-sarjan kaksimoottorisen lentokoneen, jonka kummankin moottorin toimintaa valvoo 5000 sensoria ja ne generoivat esimerkiksi 12 tunnin lennon aikana noin 844 teratavua mittausdataa. Tällä hetkellä useimmissa lentokonemoottoreissa on yleensä enintään 250 sensoria, joten mahdollisuudet valvoa moottorin toimintaa ovat aivan toista luokkaa. Tässä tapauksessa tavoitteena on tekoälyn ja kerätyn datan avulla ennustaa ja säätää moottorin toimintaa parhaan suorituskyvyn ja polttoainetalouden saavuttamiseksi. (Aviation Week 2016.)

Yhdysvalloissa Louisvillen kaupungissa on jaettu yli tuhannelle astmaa sairastavalle ihmiselle pienellä GPS-paikantimen sisältävällä sensorilla varustettu astmapiippu. Huono ilmanlaatu aiheuttaa hengitysongelmia varsinkin astmapotilaille, joten piipun käytöstä

kerättyä dataa analysoimalla on saatu selvitettyä paikkoja, joissa esiintyy ilmansaasteita. (The Wall Street Journal 2017.)

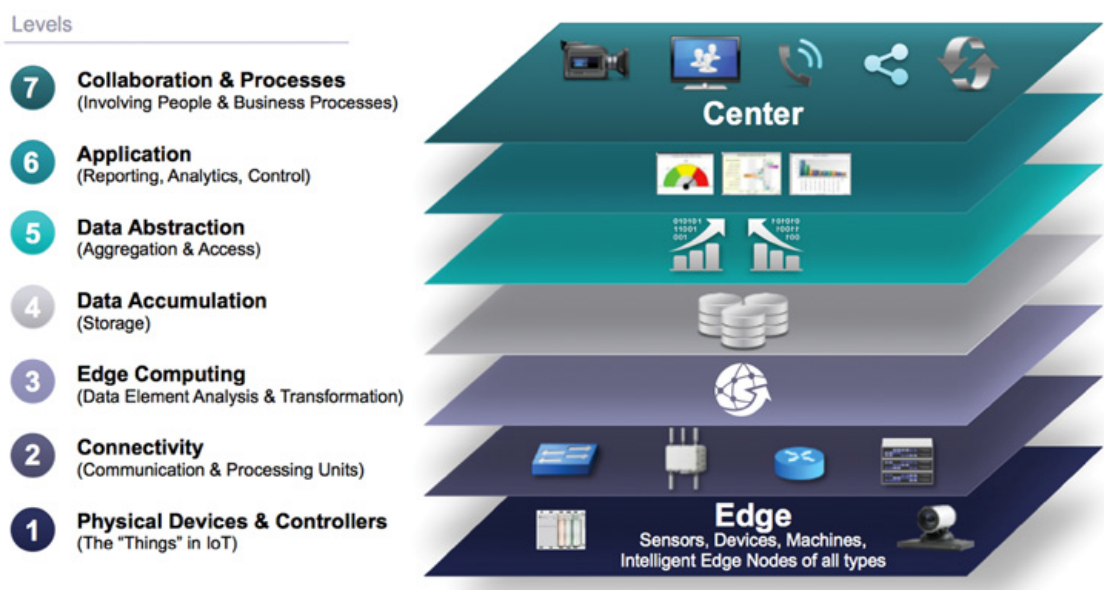
Chilessä Santiago de Chilen lentokenttä hyödyntää esineiden internetiä terminaalirakennusten olosuhteiden valvonnassa. Sensorijärjestelmä kerää tietoa esimerkiksi lämpötilasta ja kosteudesta, odotusajoista, matkatavaroiden käsittelyajoista ja tilojen käyttöasteesta. Tietoja analysoimalla pyritään energiasäästöihin sekä parantamaan matkustajien asiakaskokemusta. (Libelium 2016.)

Teknologiayhtiö Withings on yhteistyössä kosmetiikkavalmistajien kanssa kehittänyt verkkoon liitettävän hiusharjan. Harja kerää käytön aikana tietoja harjan liikkeistä ja harjausäänistä, jotka siirretään verkon kautta matkapuhelimeen. Tiedot yhdistetään säätiöjen kanssa ja näin pyritään parantamaan käyttäjän harjausrutiineja. (Wired 2017.)

3 IOT-JÄRJESTELMÄN ARKKITEHTUURI

3.1 Yleinen arkkitehtuuri

IoT-järjestelmälle asetettujen vaatimusten täyttäminen vaatii järjestelmän huolellista suunnittelua sen kaikilla osa-alueilla ja ensimmäinen vaihe on tarvittavan arkkitehtuurin ymmärtäminen ja suunnittelu. IoT World Forum on määritellyt referenssiarkkitehtuurin (kuva 2), jonka avulla IoT-järjestelmän toimintoja ja tasoja voidaan hahmottaa.



KUVA 2. IoT referenssiarkkitehtuuri (IoT World Forum)

Mallissa on seitsemän tasoa, joista jokaisella on oma funktionsa. Malli ei aseta rajoituksia komponenttien fyysiselle sijoittumiselle eikä eri tasoilla tapahtuvan tiedon prosessoinnin tasoa. Toisin sanoen, järjestelmän laitteet voivat sijaita eri puolilla maapalloa ja tietoa voidaan käsitellä millä tasolla tahansa niin monimutkaisella tavalla kuin on tarpeellista. Lisäksi mallissa informaatio liikkuu kumpaankin suuntaan. Laitteiden keräämä data välittyy tasolta 1 ylöspäin tasolle 7 ja laitteiden ohjaamiseen tarvittava data liikkuu tasolta 7 tasolle 1. (IoT World Forum 2014.)

Tasolla 1 sijaitsevat järjestelmän laitteet, sensorit ja aktuaattorit. IoT-järjestelmän tietojen keräys ja ympäristöön aktuaattorien avulla tapahtuva vaikuttaminen tapahtuu tällä tasolla.

Taso 2 sisältää verkkolaitteet joiden välityksellä tason 1 laitteet kytkeytyvät verkkoon, toisiinsa ja tason 3 toimintoihin. Lisäksi tällä tasolla voidaan tehdä tarvittavia protokollamuunnoksia ja tietoturvaan liittyviä tarkistuksia ja toimenpiteitä.

Tasolla 3 voidaan laitteiden keräämää dataa ensimmäisen kerran suodattaa tai aggregoida, jotta ylemmille tasoille jatkokäsiteltäväksi välittyvän datan määrää saadaan vähennettyä.

Tasolla 4 tallennetaan data valitulla tavalla esimerkiksi tietokantaan tai levyjärjestelmään järjestelmän vaatimusten mukaisesti. Tallennettu data on tämän jälkeen muiden ylempien tasojen saavutettavissa ja käytettävissä.

Taso 5 huolehtii datan abstraktoinnista eli järjestelmän eri lähteisiin tallennettuun dataan luodaan yhtenäiset näkymät, joita ylemmän tason sovellusten on helppo käyttää. Tasolla varmistetaan lisäksi myös datan yhtenäisyys ja suojataan se järjestelmän autentikointi- ja autorisointivaatimusten mukaisesti.

Tasolla 6 sijaitsevat sovellukset, joilla esimerkiksi analysoidaan ja visualisoidaan IoT-laitteiden keräämää ja alemmien tasojen esiprosessoimaan dataa.

Tasolla 7 ovat esineiden internetiin liittyvät prosessit ja ihmiset eli tällä tasolla hyödynnetään järjestelmän eri tasoilla tapahtuneen prosessoinnin lopputulosta. Laitteiden keräämästä datasta jalostettua tietoa voidaan käyttää esimerkiksi liiketoimintaprosessien apuna tai muulla tavalla, joka hyödyttää järjestelmän käyttäjiä.

Mallin kaksisuuntaista liikennettä hyödynnetään tasolla 1 sijaitsevien laitteiden ja aktuaattorien ohjaamiseen, valvontaan ja ohjelmistojen päivitykseen. Järjestelmään liitettyjen laitteiden määrä saattaa olla miljoonia ja ne voivat sijaita ympäri maapalloa vaikeapääsyisissä paikoissa, joten mikäli niihin ei päästä käsiksi ilman fyysistä kontaktia, järjestelmän ylläpito ja sitä kautta koko toiminta saattaa vaarantua.

IoT World Forumin referenssiarkkitehtuuri ei ole ainoa olemassa oleva malli, eikä se varmasti sovi eikä sitä ole edes tarpeellista kirjaimellisesti noudattaa kaikissa järjestelmissä. Malli ei asetakaan tiukkoja rajoituksia, vaan sitä voidaan soveltaa kulloistenkin tarpeiden mukaan. Esimerkiksi taso 3, jolla voidaan dataa suodattaa ei ole tarpeellinen, mikäli laitteiden tai niiden keräämän datan määrä on vähäinen.

3.2 Sensorit, laitteet, ja aktuaattorit

Esineiden internetistä puhuttaessa on hyvä ymmärtää IoT-maailman peruskäsitteet eli termit laite, sensori ja aktuaari.

Sensori on yksinkertainen komponentti, joka kykenee mittaamaan ja havainnoimaan jotakin fyysisen maailman tapahtumaa. Se voi esimerkiksi mitata lämpötilaa, ilmanpainetta, kiihtyvyyttä tai vaikkapa jotakin biologista elementtiä, kuten pulssia. Sensorissa ei tyypillisesti itsessään ole älykkyyttä tai kykyä kytkeytyä verkkoon, se vain muuttaa jonkin fysikaalisen suureen digitaaliseen muotoon.

Laitteella tarkoitetaan useista komponenteista ja sensoreista koottua älykästä kokonaisuutta, joka pystyy prosessoimaan ja välittämään sensoreiden mittaamaa informaatiota eteenpäin. Laitteissa on lisäksi kommunikaation mahdollistava komponentti, jonka avulla laite pystytään kytkemään joko johonkin langattomaan tai langalliseen verkkoon. Laitteet ovat esineiden internetin oleellisia rakennuskomponentteja, ilman laitteita IoT-järjestelmien rakentaminen ei ole mahdollista. Laitteiden fyysiset ominaisuudet kehittyvät kovaa tahtia, koko ja virrankulutus pienenevät, kommunikaatiomahdollisuudet paranevat ja lisäksi datan prosessointikyky kasvaa. Tämä kehitys on mahdollistanut esineiden internetin leviämisen uusille sovellusalueille.

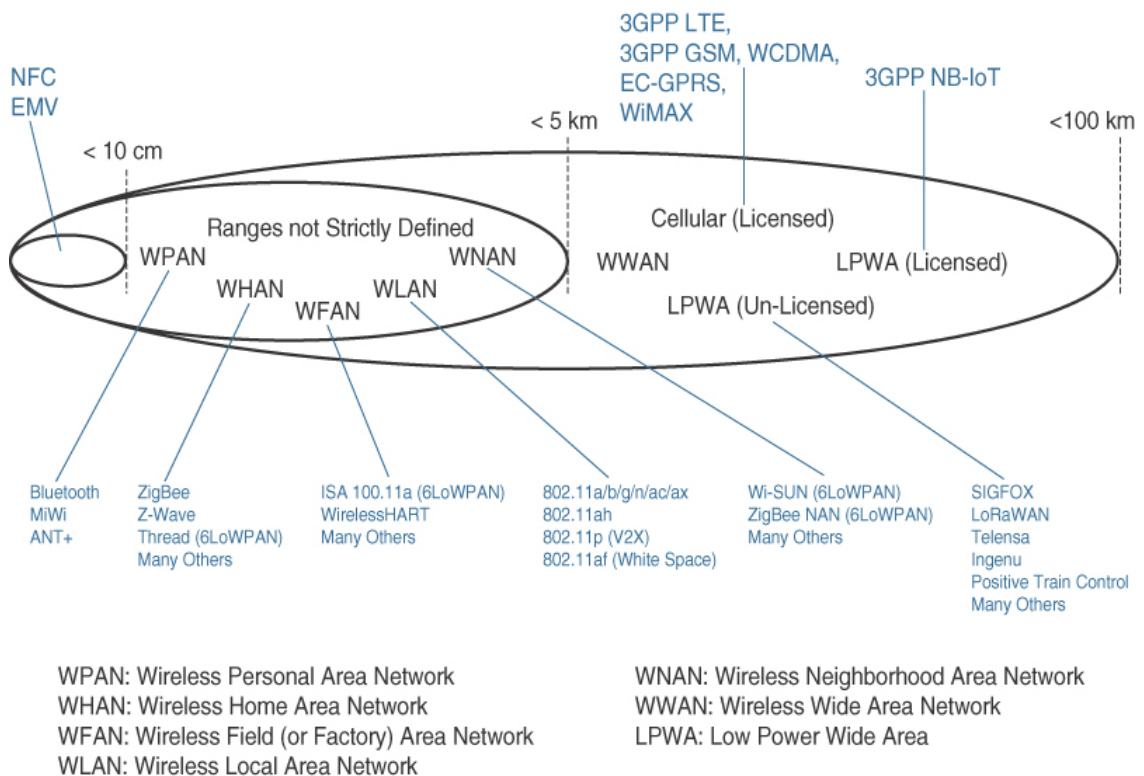
Aktuaattorilla tarkoitetaan komponenttia tai mekanismia, joka pystyy muuttamaan digitaalisen tiedon liikkeeksi tai toiminnaksi, toisin sanoen se pystyy vaikuttamaan ympäristöönsä jollakin tavalla. Aktuaattori voi esimerkiksi sammuttaa virran laitteesta, sulkea venttiilin tai sytyttää valot. IoT-järjestelmässä aktuaattoreita ohjataan sensoreiden keräämän ja IoT-järjestelmän kontrollitasolla analysoidun informaation perusteella.

Lisäksi usein puhutaan älykkäistä laitteista tai tuotteista, joka täytyy erottaa yllä määritellyistä sensoreita sisältävästä laitteesta. Älykäs tuote on esineiden internetissä varsinainen esine, jonka toimintaa seurataan ja mitataan ja joka on kytketty verkkoon. Se voi olla esimerkiksi jääkaappi, teollisuuden tuotantokone tai vaikkapa henkilöauto. Älykäs tuote sisältää useita sensoreilla varustettuja laitteita.

3.3 Verkkoteknologiat

IoT-järjestelmään voidaan liittää laitteita monilla erilaisilla teknologioilla. Verkkoteknologioiden valintaan vaikuttavat monet seikat, kuten järjestelmän asettamat vaatimukset ja käyttötarkoitus. Huomioitavia asioita ovat esimerkiksi vaadittava kantomatka, virrankulutus, kaistanleveys ja turvallisuus. Tyypillisesti järjestelmään voidaan liittää laitteita useilla teknologioilla, harvoin laitekanta sisältää vain yhtä yhteysteknologiaa tukevia laitteita.

Eri yhteysteknologioiden kantomatka vaihtelee muutamista senttimetreistä kymmeniin kilometreihin (kuva 3). Puettaviin sovelluksiin voidaan valita lyhyttä kantamaa kuten esimerkiksi Bluetoothia tukevia laitteita, kaukana maastossa toimivien kohteiden liittämiseen vaikkapa LTE-teknologiaa käyttäviä laitteita. Pieni virrankulutus on laitteiden keskeinen vaatimus koska ne eivät useinkaan ole kytkettyjä kiinteään virtalähteeseen, vaan ne on varustettu paristolla tai akulla, joten mahdollisimman pitkän toiminta-ajan mahdollistaminen on erittäin tärkeää. Yhteysteknologioiden virrankulutus vaihtelee huomattavasti, lähes poikkeuksetta pidempi kantavuus tarkoittaa myös suurempaa virrankulutusta.

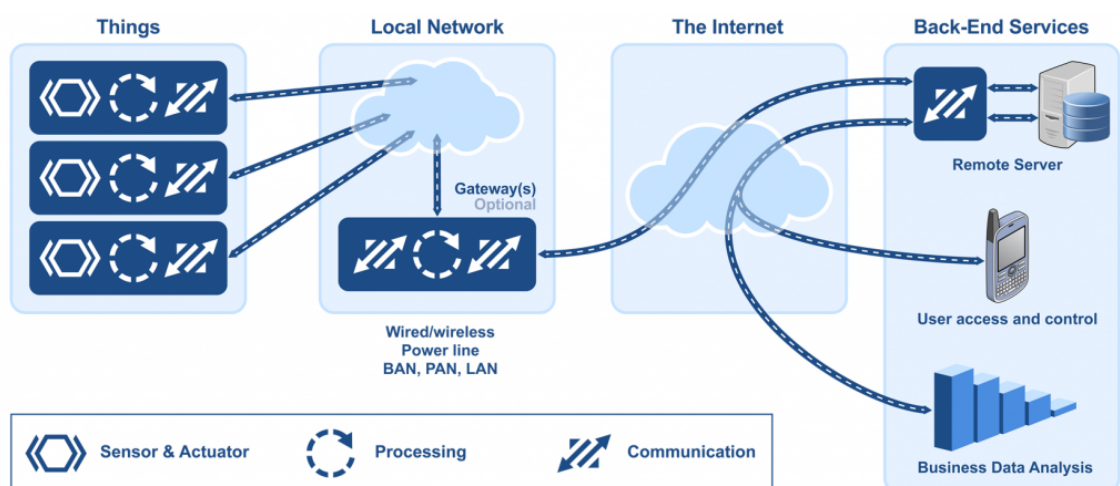


KUVA 3. Yhteysteknologioiden toimintamatkat (Hanes D., Salgueiro G. & Barton R. 2017)

Laitteiden keräämän ja välittämän datan määrä vaikuttaa myös käytettävien yhteysteknologioiden valintaan. Laitteet saattavat lähettää sensoreilta keräämäänsä dataa joko jatkuvasti tai tietyn väliajoin purskeina, jolloin myös verkon täytyy pystyä data ongelmitta välittämään. Lisäksi yhteysteknologian täytyy soveltua kaksisuuntaiseen liikenteeseen, mikäli järjestelmää halutaan käyttää kytkettyjen laitteiden tai aktuaattoreiden valvontaan tai ohjaamiseen.

Verkon turvallisuus on erittäin tärkeä ominaisuus, jonka täytyy olla kunnossa. Laitteiden keräämä informaatio usein sisältää kriittistä tietoa, joka ei saa päätyä sivullisten käsiin. Verkkoon liittävät laitteet täytyy myös suojata, jotta niiden kautta ei ulkopuolinen pääse vaikuttamaan järjestelmän toimintaan. Verkko suojataan tyypillisesti autentikoinnilla ja siinä kulkeva tietoliikenne salataan.

Verkkoon liitettävät laitteet voidaan myös jakaa kahteen perustyyppiin. Laite voi kyetä kommunikoimaan suoraan internet-verkon ja siihen liitettyjen toisten laitteiden kanssa tai se voi vaatia välilaitteen, joka on kytketty internet-verkkoon ja tekee välissä tarvittavan protollamuunnoksen. Internet-kerroksessa käytetään yleensä IPv6 tai 6LoWPAN teknologioita, jotka mahdollistavat massiivisen laitemäärän kytkemisen verkkoon. Koska suuri määrä IoT-laitteita tarvitsee osoitteen, nykyään käytössä olevaa rajoitetun osoiteavaruuden tarjoavaa IPv4:n käyttämistä on syytä välttää. Kuvassa 4 nähdään tyypillinen IoT-verkon rakenne.



KUVA 4. IoT-verkko (Micrium, Devices & Networks)

3.4 Sovellustason viestiprotokollat

Sovellustasolla käytettyjä yleisiä protokollia ovat muun muassa HTTP/HTTPS, MQTT ja XMPP. Laitteet kommunikoivat mainituilla protokollilla joko suoraan IoT-järjestelmän tarjoaman rajapinnan tai sen protokollamuunnos-palvelun kanssa. Jälkimmäinen mahdollistaa omaa yhteysprotokollaa käyttävien laitteiden liittämisen. Kuvassa 5 nähdään eri protokollien ominaisuuksia.

Kommunikointi HTTP(S)-protokollan välityksellä tapahtuu tyypillisesti välittämällä JSON-muotoisia sanomia REST-rajapinnan kautta. Etuina voidaan pitää JSON sanomien helppoa generointi ja lähetystä sekä laajaa tukea laiterintamalla. Viestien perille menoa ei kuitenkaan pystytä takaamaan pelkän HTTP(S) –protokollan avulla, vaan järjestelmään täytyy rakentaa ylimääräisiä toimintoja, joilla saadaan toimitus varmistettua.

MQTT on julkaise-tilaa -protokolla, jota pidetään erityisen sopivana IoT-järjestelmien käyttöön, koska se on suunniteltu kevyeksi ja pienet resurssivaatimukset omaavaksi protokollaksi. Protokollaan on myös rakennettu QoS –ominaisuus, jolla viestien toimitusta voidaan ohjata varmistaen näin niiden perille meno asiakkaalta (laite) vastaanottajalle (MQTT 2017).

XMPP on avoin XML-pohjainen protokolla, joka on suunniteltu reaaliaikaiseen viestien välitykseen. Lisäksi XMPP-viestejä voidaan laajentaa järjestelmän vaatimusten mukaan. XMPP edellyttää myös verkkoliikenteen TLS-salausta ja SASL –tunnistautumista.

	DDS	AMQP	CoAP	MQTT	REST / HTTP	XMPP
TRANSPORT	UDP/IP (unicast + multicast) TCP/IP	TCP/IP	UDP/IP	TCP/IP	TCP/IP	TCP/IP
INTERACTION MODEL	Publish-and-Subscribe, Request-Reply	Point-to-Point Message Exchange	Request-Reply (REST)	Publish-and-Subscribe	Request-Reply	Point-to-Point Message Exchange
SCOPE	Device-to-Device Device-to-Cloud Cloud-to-Cloud	Device-to-Device Device-to-Cloud Cloud-to-Cloud	Device-to-Device	Device-to-Cloud Cloud-to-Cloud	Device-to-Cloud Cloud-to-Cloud	Device-to-Cloud Cloud-to-Cloud
AUTOMATIC DISCOVERY	✓	-	✓	-	-	-
CONTENT AWARENESS	Content-based Routing Queries	-	-	-	-	-
QoS	Extensive (20+)	Limited	Limited	Limited	-	-
INTEROPERABILITY LEVEL	Semantic	Structural	Semantic	Foundational	Semantic	Structural
SECURITY	TLS, DTLS, DDS Security	TLS + SASL	DTLS	TLS	HTTPS	TLS + SASL
DATA PRIORITIZATION	Transport Priorities	-	-	-	-	-
FAULT TOLERANCE	Decentralized	Implementation-Specific	Decentralized	Broker is SPoF	Server is SPoF	Server is SPoF

KUVA 5. Viestiprotokollien ominaisuuksia (Prismtech)

4 IOT-ALUSTAT

4.1 Mikä on IoT-alusta

Esineiden internet ei ole yksi teknologia, vaan se on kokonaisuus, joka muodostuu useista eri teknologioista, laitteista ja ohjelmistoista. Sensorit ja laitteet tukevat erilaisia yhteystapoja ja niiden välittämän informaation sisältö ja muoto vaihtelevat sekä lisäksi informaation tallennus- ja käsittelytarpeet saattavat suuresti vaihdella käyttötapauksittain. IoT-järjestelmä on aivan mahdollista rakentaa suhteellisen pienellä panostuksella käyttämällä vaikkapa pelkästään vapaasti saatavilla olevia avoimen lähdekoodin tuotteita, mikäli kyseessä on erittäin yksinkertainen käyttötapaus ja liitettyjen laitteiden ja kerätyn datan määrä on vaatimaton. Tarpeiden ja vaatimusten kasvaessa ja järjestelmän tullessa yhä monimutkaisemmaksi, saattaa valmiin IoT-alustan käyttäminen tulla kyseeseen.

Perry M. (2016) määrittelee IoT-alustan järjestelmäksi, joka helpottaa kommunikaatiota, tietovirtojen hallintaa ja sovellusten toimintaa ja yhdistää laitteet, sovellukset ja ihmiset tietovarastoihin ja hallintajärjestelmiin. Järjestelmän perusosia ovat tyypillisesti erilaiset rajapinnat laitteiden kytkeytymistä ja hallintaa varten, informaation tallennuksen ja prosessoinnin suorittavat komponentit, työkalut datan visualisointiin sekä rajapinnat ulkopuolisten sovellusten ja järjestelmien liittämiseen.

IoT-alustan tarjoamia ominaisuuksia käyttäen järjestelmän rakentamisen painopiste saadaan siirrettyä perustoiminnoista kuten laitteiden liitettävyydestä ja tietojen tallennuksesta enemmän varsinaisten liiketoimintavaatimusten ja -logiikan toteuttamiseen. Perustoimintojen suunnittelu ja toteutus on aikaa vievää ja kallista, eikä sitä useinkaan kannata tehdä itse alusta alkaen, mikäli siihen ei ole perusteltua tarvetta.

IoT-alustoja on tänä päivänä saatavilla hyvin monen taseisia ja moniin erilaisiin tarpeisiin. Osa alustoista on toteutettu avoimen lähdekoodin periaatteella ja lisäksi kaupallisia tuotteita löytyy monenlaisilla hinnoittelumalleilla.

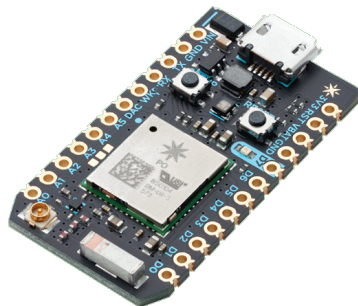
4.2 IoT-alustojen tyypit

Markkinoilta tällä hetkellä löytyvien IoT-alustojen kirjo on suuri. Alustoja on monenlaisia eri käyttötarpeisiin ja ne voidaan jaotella kolmeen pääluokkaan.

Ohjelmistoalustat ovat joko omaan datakeskukseen tai pilviympäristön palvelimille asennettavia ratkaisuja. Ne sisältävät IoT-järjestelmän komponentteja ja sovelluksia, jotka tarjoavat ohjelmistorajapinnat laitteiden kytkemistä varten, tiedon tallentamiseen ja käsittelyyn vaadittavat komponentit sekä työkalut tiedon analysointiin ja visualisointiin. Lisäksi ulkopuolisten sovellusten liittämiseen tarjotaan yleensä erilaisia rajapintoja. Esimerkkejä ohjelmistoalustoista ovat Kaa IoT Platform ja ThingWorx.

Pilvialustat tarjoavat ohjelmistoalustan toimintojen lisäksi infrastruktuurin. Pakettiin kuuluu tarvittavien resurssien mukaan skaalautuva palvelu, jossa toimittaja vastaa verkko- ja laiteresurssien toiminnasta ohjelmistoalustan lisäksi. Käyttäjän tehtävänä on sovittaa haluamansa IoT-laitteet ja toteuttaa varsinaiset sovellukset alustan tarjoamilla työkaluilla ja palveluilla. Tunnettuja esimerkkejä pilvialuista ovat Microsoft Azure IoT Hub ja Amazon IoT Platform.

Laitealustat tarjoavat valmiita komponentteja varsinaisten IoT-laitteiden rakentamiseen. Komponentteja ovat esimerkiksi piirilevyt ja –sarjat ja niiden käyttämät oheislaitteet kuten erilaiset sensorit. Jotkin laitealustojen varmistajat tarjoavat lisäksi myös ohjelmistotai pilvialustoja laitteiden ohessa. Laitealustoja ovat Raspberry Pi, Particle Photon (kuva 6) ja ARM mbed.



KUVA 6. Particle Photon piirilevy (Particle)

4.3 IoT-alustan vaatimukset

IoT-järjestelmään liitettyjen laitteiden ja kerätyn tiedon määrä saattaa olla valtava ja sen käsittely vaatii paljon resursseja ja monenlaisia toimintoja. Järjestelmän täytyy pystyä vastaamaan puhtaasti teknisten vaatimusten lisäksi myös moniin muihin tarpeisiin. Keskeisimpiä IoT-alustaan kohdistuvia vaatimuksia ovat muun muassa seuraavat.

Alustan täytyy olla skaalautuva. Miljoonien kytkettyjen laitteiden generoima verkkoliikenne sekä välitetyn ja tallennetun datan määrä saattaa olla massiivinen. Datan tallennukseen, analysointiin ja erilaisten järjestelmään liitettyjen sovellusten palvelemiseen tarvitaan skaalautuvia resursseja. (Hanes D., Salgueiro G. & Barton R. 2017.)

Alusta täytyy olla yhteensopiva ja tukea useita verkkoprotokollia laitteiden liittämiseksi sekä erilaisia rajapintoja ja arkkitehtuureja, joilla saadaan toteutettua varsinaiset liiketoimintavaatimukset ja liittymät muihin järjestelmiin. (Hanes D., Salgueiro G. & Barton R. 2017.)

Alustan täytyy olla turvallinen. Järjestelmässä täytyy olla menetelmät laitteiden turvalliseen liittämiseen ja hallintaan, hakkeroidun laitteen liittäminen saattaa vaarantaa koko järjestelmän tietoturvan. (Hanes D., Salgueiro G. & Barton R. 2017.)

Alustan täytyy huomioida yksityisyys. Laitteiden keräämä data saattaa sisältää tietoa henkilöistä tai organisaatioista eikä sitä saa päästää vuotamaan väärin käsiin. (Hanes D., Salgueiro G. & Barton R. 2017.)

Alustan täytyy kyetä hallinnoimaan liitettyjä laitteita. Laitteiden toiminnan konfigurointi, valvonta ja ohjelmistojen päivitys ovat olennainen osa toimivaa järjestelmää. (Perry M. 2016.)

Alustan täytyy tarjota joustavat kaksisuuntaiset yhteydet eikä se saa olla ainoastaan tietovarasto, johon tallennetaan laitteiden keräämä informaatio. (Perry M. 2016.)

Alustan täytyy tarjota riittävät ominaisuudet data analysointiin. Laitteiden keräämä ja järjestelmään tallennettu informaatio on koko toiminnan ydin, mikäli sitä ei pystytä riittävän monipuolisesti käsittelemään, järjestelmä menettää merkityksensä. (Perry M. 2016.)

5 ESIMERKKEJÄ IOT-ALUSTOISTA

5.1 Kaupalliset tuotteet

IoT-alustojen tarjonta on huomattavan suuri ja odotettavissa on lisää toimijoita esineiden internetin markkinoiden yhä kasvaessa. Tällä hetkellä markkinoilta löytyy yli 450 yri-tystä, joiden valikoimassa on IoT-alusta jossakin muodossa (IoT Analytics 2017). Alus-
tamarkkinoiden rahallinen arvo on jo nyt merkittävä ja pelkästään pilvialustojen markki-
noiden arvioidaan kohoavan jopa yli seitsemään miljardiin dollariin vuoteen 2021 men-
nessä (MarketsandMarkets 2016).

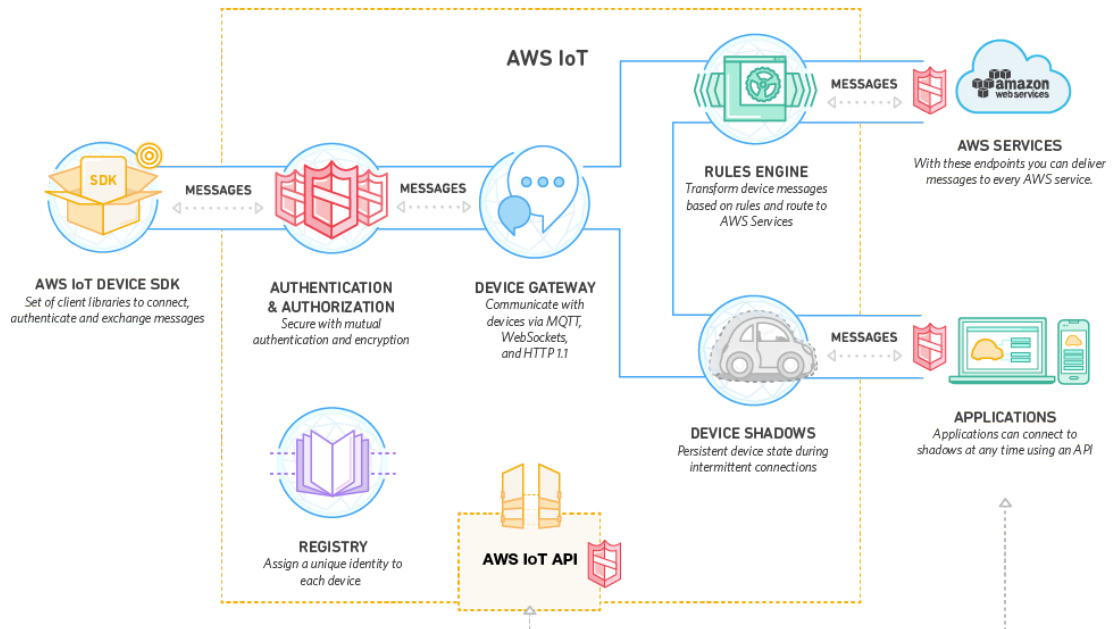
Lähes kaikki merkittävät IT-alan toimijat ovat tavalla tai toisella mukana esineiden inter-
netin kehittämisessä ja tarjoavat joko omia alustoja tai ovat mukana erilaisissa allians-
seissa, jotka kehittävät esineiden internetin alustoja ja standardeja.

5.1.1 AWS IoT Platform

Amazon julkaisi IoT-pilvialustansa vuoden 2015 loppupuolella (Business Wire 2015). AWS IoT Platform mahdollistaa IoT-laitteiden liittämisen Amazonin pilvipalveluihin ja näin tarjoaa tavan IoT-järjestelmän rakentamiseksi. Alusta tarjoaa lisäksi skaalautuvan infrastruktuurin ja käytön mukaan muuttuvan hinnoittelun. Hinnoittelu perustuu lähetet-
tyjen viestien lukumäärään. Palveluun on mahdollista tutustua ilmaiseksi vuoden ajan, jolloin palvelun käyttö on rajoitettu 250 000 lähetettyyn viestiin kuukaudessa (Amazon IoT Platform 2017).

Amazon ei valmista varsinaisia IoT-laitteita, mutta tarjoaa yhteistyökumppaneiden lait-
teita, joiden avulla järjestelmän kehityksessä pääsee nopeasti alkuun. Laitteisiin on tar-
jolla ohjelmistokehityspaketti C/JavaScript/Java/Python –kielille, jonka avulla laitteet
voidaan helposti liittää IoT-alustaan tuettuja MQTT-, HTTP- tai WebSocket –protokollia
käyttäen.

Alustassa ei itsessään ole datan visualisointiin tarvittavia komponentteja, ideana on ohjata sääntömoottorin (Rules Engine) prosessoima data muihin Amazonin pilvipalveluihin, kuten esimerkiksi visualisointiin tarkoitettuun Amazon QuickSight-palveluun. Kuvassa 7 on esitetty Amazon IoT-alustan tarjoamia palveluita,

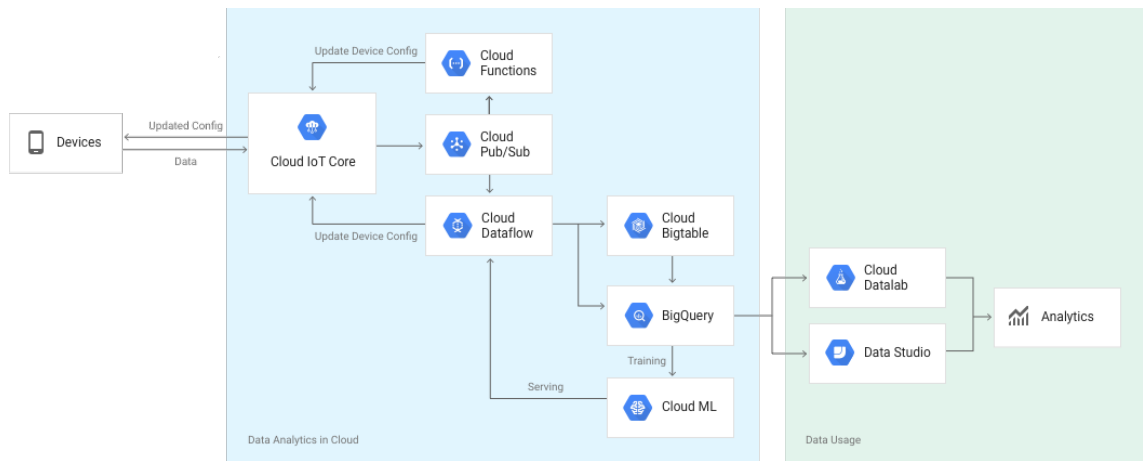


KUVA 7. Amazon IoT-alusta (Amazon IoT Platform)

5.1.2 Google Cloud IoT Core

Google julkaisi IoT-palvelunsa vuoden 2017 puolivälissä. Kuten Amazon, myös Googlen Cloud IoT Core –tuote tarjoaa väylän IoT-laitteiden liittämiseen Googlen pilvipalveluun, jonka muilla tuotteilla on mahdollista rakentaa halutut toiminnot (kuva 8). Palvelun hinnoittelu perustuu siirrettyyn datamäärään, palvelua on mahdollista kokeilla ilmaiseksi edellyttäen, että datamäärä jää kuukaudessa alle 250 megatavun. Cloud IoT Core tukee MQTT ja HTTP protokollia, joten kaikki näitä tukevat laitteet on helppo liittää palveluun (Google Cloud IoT 2017).

Kuten kilpaileva toimittaja Amazon, myös Google on yhteistyössä laite- ja ohjelmistovalmistajien kanssa, joiden kautta on saatavissa järjestelmään tutustumista ja kehittämistä varten IoT-laitteita ja sensoreita sekä Googlen alustaan integroituvia ohjelmistoja, joita voidaan käyttää tiedon analysointiin ja visualisointiin.



KUVA 8. Google Cloud IoT (Google Cloud IoT)

5.1.3 Arm Mbed IoT Platform

Puolijohdevalmistaja ja ohjelmistotalo Arm valmistaa Cortex-prosessoriperheen ympärille rakennettuja piirilevyjä ja komponentteja. Näistä rakennetut laitteet käyttävät avoimen lähdekoodin sulautettua Mbed OS käyttöjärjestelmää, johon on sisäänrakennettu monia esineiden internetin vaatimia ominaisuuksia, kuten esimerkiksi tuki Bluetooth, 6LoWPAN- ja WiFi-yhteystechnologioille sekä erilaisia tietoturvaominaisuuksia. Laittealustalle tapahtuvan ohjelmistokehityksen tueksi Arm tarjoaa lisäksi Mbed OS:lle suunnitellun kehitysympäristön.

Mbed ekosysteemiin liittyy myös Mbed Cloud –pilvipalvelu, joka on suunniteltu erityisesti Mbed OS –laitteisiin pohjautuvan IoT-järjestelmän alustaksi. Palvelu on tarkoitus avata vuoden 2017 aikana ja se korvanee aiemmat Armin IoT-pilvipalvelut (Arm Mbed 2017).

5.2 Open Source –tuotteet

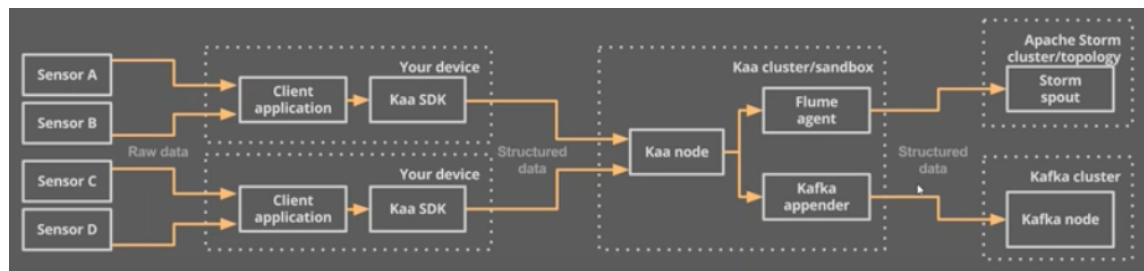
Esineiden internetiin liittyviä avoimen lähdekoodin tuotteita on tarjolla runsaasti, markkinoilta löytyy sekä ohjelmisto/pilvi-alustoja että IoT-laitteiden rakentamiseen tarkoitettuja laitealustoja. Monista tuotteista on saatavilla ilmaisen käytön mahdollistava versio sekä maksullinen versio, jonka mukana tarjotaan laajennettuja ominaisuuksia, suurempia viestimääriä tai isompaa tallennustilaa.

5.2.1 Kaa IoT Platform

Kaa IoT Platform on avoimen lähdekoodin IoT-alusta. Kaa on sovellusalusta, joka tarjoaa rajapinnan datan tallentamiseen IoT-laitteista ja rajapinnat datan käyttämiseen sovelluksista ja muista järjestelmistä käsin. Kaa toteuttaa joukon yhteysprotokollia sekä laitteiden että sovellusten käyttöön ja tarvittaessa on mahdollista luoda uusia kustomoituja yhteystapoja. Kuvassa 9 esitetään Kaa-alustan perusarkkitehtuuri.

Kaa SDK:n avulla toteutetaan asiakassovellus IoT-laitteisiin, jonka avulla lähetetään kerätty data Kaa-taustajärjestelmään, jossa sitä voidaan prosessoida sovelluksissa ja muissa järjestelmissä käytettäväksi. Kaa SDK on saatavilla Java-, C/C++- ja Objective-C -ohjelmointikielillä ja se tukee laajaa joukkoa laitealustoja (muuan muassa RaspberryPi, Intel Edison, Texas Instruments). Kaa SDK on käytännössä mahdollista sovittaa kaikille laite- ja ohjelmistoalustoille, jotka tukevat yllä mainittuja ohjelmointikieliä.

Järjestelmän keskeinen osa on Kaa Server, joka huolehtii rajapintojen tarjoamisesta, datan tallentamisesta sekä tarjoaa työkalut järjestelmän konfigurointiin ja monitorointiin. Kaa IoT Platform voidaan asentaa omaan infrastruktuuriin tai pilvipalveluihin, kuten esimerkiksi Amazon-palveluun (Kaa 2017).



KUVA 9. Kaa IoT Platform arkkitehtuuri (Kaa IoT Platform)

5.2.2 BeagleBoard

BeagleBoard on avoin Linux-pohjainen laitealusta, jonka mahdollistaa nopean IoT-laitteiden prototyyppien ja niiden sovellusten kehityksen (BeagleBoard 2017). Alustaan perustuvia laitteita voi ostaa valmiina tai sellaisen voi rakentaa itse. Avoimeen laitealustaan perustuvien tuotteiden suunnitelmat ja määrittelyt ovat vapaasti saatavilla ja niitä voidaan

käyttää omien tuotteiden suunnittelun apuna sekä niihin perustuvia tuotteita voidaan vapaasti valmistaa ja myydä. Yleensä laitteet on myös suunniteltu vapaasti markkinoilta saatavia komponentteja käyttäen, jolloin laitteiden rakentaminen onnistuu mahdollisimman helpolla tavalla.

BeagleBoard-laitteissa käytetään Linux-käyttöjärjestelmää, joka mahdollistaa laajan valikoiman ohjelmistoja ja ohjelmointikieliä, joita käyttäen laitteisiin voidaan toteuttaa IoT-laitteen ominaisuuksia, kuten tietojen keräämisen liitettyjen sensoreiden avulla ja tulosten lähettämisen IoT-järjestelmään. Usean muun laitetoimittajan tavoin myös BeagleBoard on yhteistyössä muiden IoT-toimijoiden kanssa ja sen tuotteisiin perustuvia starttipaketteja on saatavilla esimerkiksi Amazon IoT-palvelun kautta.

6 THINGSBOARD

6.1 Esittely

ThingsBoard on avoimen lähdekoodin IoT-ohjelmistoalusta, joka tarjoaa suhteellisen yksinkertaisen ja helpon tavan tutustua esineiden internetissä toimivan järjestelmän tekniiseen toteutukseen. Alusta tarjoaa työkalut laitteiden lähettämän tiedon keräämiseen ja tallentamiseen, tiedon prosessointiin sekä erityisenä vahvuutena monipuoliset tiedon visualisointiin keskittyvät toiminnot. Alusta mahdollistaa myös siihen liitettyjen laitteiden ohjaamisen sekä tarjoaa rajapinnat sekä olemassa olevien toimintojen laajentamiseen, että uusien toteuttamiseen. Myös ulkopuolisten sovellusten liittäminen järjestelmään on mahdollista.

Useisiin muihin alustoihin verrattuna ThingsBoard paketoi kaikki perustoiminnot yhdeksi kokonaisuudeksi, joka on helppo ottaa käyttöön. ThingsBoard on lisensoitu Apache 2.0 –lisenssillä, joten sitä on mahdollista käyttää myös kaupallisiin tarkoituksiin, lisäksi siihen on saatavissa maksullista koulutusta ja tukea (ThingsBoard 2017).

6.1.1 Laitevaatimukset

ThingsBoard-järjestelmän laitevaatimukset riippuvat useista eri seikoista, kuten esimerkiksi järjestelmän aiotusta käyttötarkoituksesta, liitettyjen laitteiden ja niiden generoiman liikenteen määrästä sekä toteutetuista visualisointi-näkymistä ja järjestelmän ulkopuolisista sovelluksista.

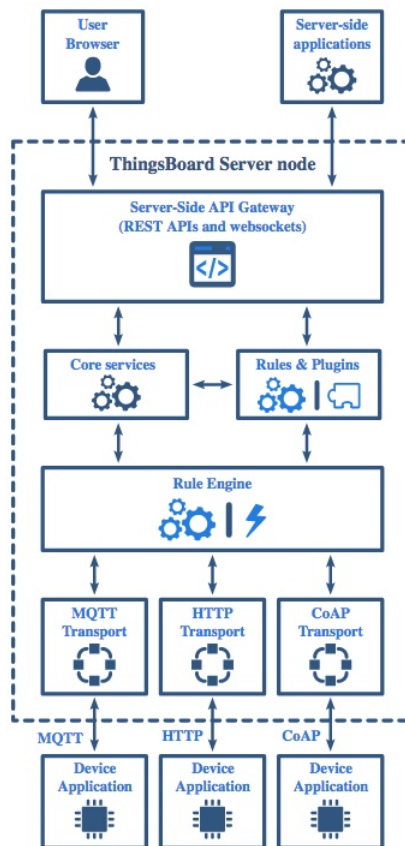
Järjestelmä voidaan asentaa Linux- ja Windows -laitteisiin ja jopa Raspberry Pi –minitietokoneeseen. Linux-laitteissa vaaditaan minimissään 1 Gt RAM-muistia ja Windows-laitteissa 2 Gt, tämän lisäksi täytyy asentaa Java 8 –ajoympäristö. Koko järjestelmä voidaan tietokantoihin asentaa yhdelle laitteelle, mutta järjestelmän tarpeiden kasvaessa se voidaan klusteroida, lisäksi tuotantokäytössä on myös syytä käyttää erillisiä tietokantapalvelimia.

6.1.2 Arkkitehtuuri

Laitteiden liittämistä varten ThingsBoard tarjoaa suoraan kolme rajapintaa, jotka ovat MQTT, CoAP ja HTTP. Kunkin protokollan avulla välitetty viesti on oletusarvoisesti aina JSON-muotoinen, jossa sisältö on määritelty avain-arvo pareilla. Viestissä avain on aina merkkijono ja arvo voi olla merkkijono, boolean tai double/long –numeroarvo.

```
{"avain1" : "arvo1", "avain2" : true, "avain3" : 12.0, "avain4" : 45}
```

Tarvittaessa protokollia voidaan kustomoida ja lisäksi on mahdollista toteuttaa laajennusosa (plugin), jolla saadaan liitettyä laitteet jotka käyttävät sellaista protokollaa, jota ThingsBoard-järjestelmä ei suoraan tue. ThingsBoard-tuoteperheeseen kuuluu lisäksi IoT Gateway, jonka avulla on mahdollista liittää järjestelmään laitteita, jotka eivät tue MQTT, CoAP tai HTTP-protokollia tai ne ovat jo kytkettyjä toiseen järjestelmään, joka halutaan liittää ThingsBoard-järjestelmään esimerkiksi MQTT-protokollaa käyttäen. Kuvassa 10 on kuvattu tuotteen arkkitehtuuri.



KUVA 10. ThingsBoard arkkitehtuuri (ThingsBoard)

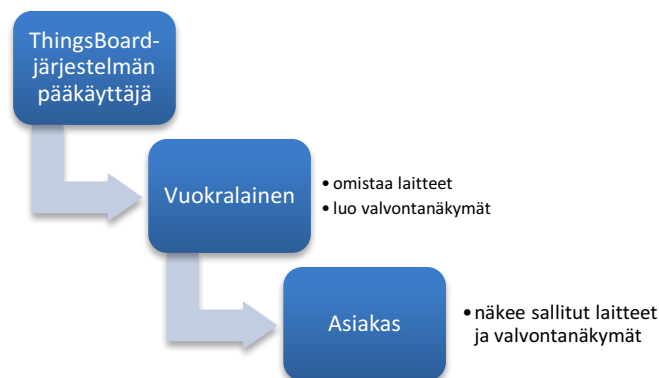
Sääntömoottori (rule engine) kostuu kahdesta osasta, säännöistä (rule) ja laajennuksista (plugin) ja se mahdollistaa järjestelmään liitettyjen laitteiden lähettämien viestien prosessoinnin. Säännön avulla analysoidaan dataa ja laajennuksen avulla suoritetaan jokin toiminto, voidaan esimerkiksi generoida hälytys jonkin raja-arvon ylityttyä tai edelleen lähettää tietyiltä laitteilta tuleva data johonkin ulkopuoliseen järjestelmään.

ThingsBoard-järjestelmän ydinpalvelut (core services) sisältävät keskeisiä toimintoja, kuten esimerkiksi järjestelmän ja laitteiden konfiguraatiodot, käyttäjätiedot ja -oikeudet sekä tiedon visualisoinnin mahdollistavat toiminnot valvontanäkymien (dashboard) ja niillä näytettävien vimpaimien (widget) luomiseen ja konfigurointiin.

Järjestelmän ulkopuoliset sovellukset voidaan liittää tarjotun REST-rajapintojen kautta. Rajapinta mahdollistaa järjestelmän eri osa-alueiden konfiguroinnin, laitteiden ja käyttäjien hallinnan sekä järjestelmään tallennetun laitteiden lähettämän datan hakemisen.

6.1.3 Käyttäjät ja laitteet

ThingsBoard-järjestelmässä käyttöoikeudet perustuvat malliin, jossa laitteet ja näkymät määritellään kuuluvaksi tietylle organisaatiolle. Järjestelmän käyttäjämalli on kolmetasoinen ja jota hallitsee pääkäyttäjä. Pääkäyttäjä luo web-pohjaisen hallintasovelluksen kautta järjestelmään yhden tai useamman vuokralaisen (tenant), joka on organisaatio joka hallinnoi omia järjestelmään liitettyjä laitteitaan. Vuokralaisen pääkäyttäjä puolestaan luo omat asiakasorganisaationsa ja näkymät, joihin antaa pääsyn asiakkaalle (kuva 11).

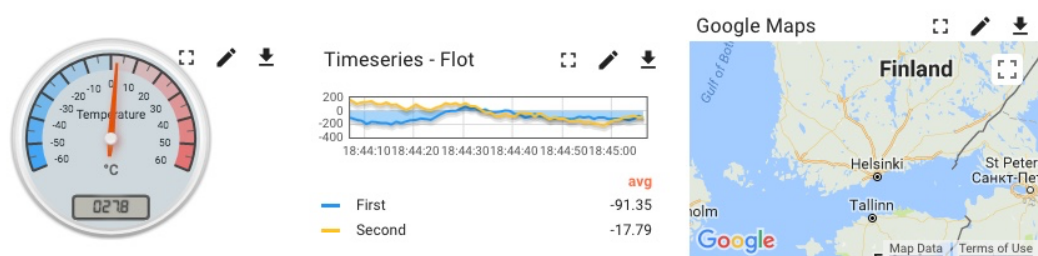


KUVA 11. Käyttäjätasot

Laitteiden liittäminen järjestelmään voidaan tehdä kahdella eri tavalla. Mikäli laitteiden määrä on vähäinen, ne voidaan rekisteröidä hallintasovelluksessa. Käytännössä tuotantojärjestelmän ollessa kyseessä ja jopa miljoonien laitteisen manuaalinen rekisteröinti ei ole mahdollista, voidaan se tehdä suoraan laitteista käsin käyttäen järjestelmän tarjoamaa REST-hallintarajapintaa.

6.1.4 Valvontanäkymä ja vimpaimet

Järjestelmän valvontanäkymiin määritellään erilaisia vimpaimia, joiden avulla laitteiden lähettämä informaatio visulisoidaan reaaliajassa. Käytettävissä olevia eri tyyppisiä vimpaimia on noin 30 kappaletta ja uusia on tarvittaessa mahdollista kehittää ja lisätä järjestelmään. Vimpaimia on monia eri tyyppisiä, analogisista mittareista karttapohjiin, kuvassa 12 on esitetty joitakin käytettävissä olevia vimpaimia.



KUVA 12. Valvontanäkymän vimpaimia

6.1.5 Järjestelmään liitettyjen laitteiden ohjaaminen

Järjestelmä mahdollistaa kaksisuuntaisen kommunikoinnin siihen liitettyjen laitteiden kanssa RPC-komentoja käyttäen. RPC-yhteys voidaan avata joko laitteesta käsin, jolloin laite lähettää pyynnön tehdä jokin toiminto palvelimen päässä jonka jälkeen laitteelle palautetaan joko vahvistus toiminnon suorittamisesta tai jokin muu sen pyytämä tieto.

Palvelimen päästä avatun yhteyden avulla voidaan ohjata tarvittaessa laitteen toimintaa tai kysyä tietoa sen sisäisestä tilasta, jota laite ei välitä tavanomaisten attribuuttien julkaisemisen yhteydessä. Esimerkiksi laitteen ohjelmistoversiota ei yleensä välitetä mittaus-tietojen mukana.

6.2 Käytännön tutustuminen järjestelmään

ThingsBoard-järjestelmän ominaisuuksiin tutustuminen onnistuu parhaiten ottamalla se käyttöön, jolloin eri toimintoja pääsee kokeilemaan käytännössä. Tarkoitusta varten asennetaan järjestelmä, lähetetään siihen simuloitua laitedataa ja konfiguroidaan valvontanäkymä, jossa data visualisoidaan.

Esimerkkiä varten asennetaan ThingsBoard-järjestelmä CentOS 7.4 Linux -palvelimelle. Asennuksessa käytetään kehitys- ja tutustumiskäyttöön tarkoitettua paketin mukana valmiina tulevaa sulautettua tietokantaa.

Asentaminen vaatii vain muutaman vaiheen. Ensin asennetaan palvelimelle ladattu Java 8 –ympäristö rpm-paketista, sen jälkeen ladataan ja asennetaan ThingsBoard-asennuspaketti ja lopuksi käynnistetään sovellus sekä avataan palvelimen paikalliseen palomuriin hallintakäyttöliittymää ja MQTT-protokollaa varten portit 8080 ja 1883. Käytetyt portit voidaan tarvittaessa määritellä järjestelmän konfiguraatiotiedostossa. Käytettyjä tietoliikenneyhteyksiä ei suojattu TLS-salauksella, oikeassa kehitys- tai tuotantoympäristössä se on kuitenkin erittäin suositeltavaa.

```
[centos@iot ~]$ sudo rpm -ivh jdk-8u151-linux-x64.rpm
[centos@iot ~]$ java -version
java version "1.8.0_151"
Java(TM) SE Runtime Environment (build 1.8.0_151-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.151-b12, mixed mode)
[centos@iot ~]$ wget https://github.com/thingsboard/thingsboard/releases/download/v1.3.1/thingsboard-1.3.1.rpm
[centos@iot ~]$ sudo rpm -Uvh thingsboard-1.3.1.rpm
[centos@iot ~]$ sudo /usr/share/thingsboard/bin/install/install.sh
[centos@iot ~]$ sudo service thingsboard start
[centos@iot ~]$ sudo firewall-cmd --add-port=8080/tcp --permanent
[centos@iot ~]$ sudo firewall-cmd --add-port=1883/tcp --permanent
[centos@iot ~]$ sudo firewall-cmd --reload
```

Asennuksen jälkeen hallintasovellukseen päästään selaimella käyttämällä osoitetta `http://<hostname tai IP>:8080`.

6.2.1 Laitesimulaattori

ThingsBoard-järjestelmään liitettyjä laitteita simuloidaan yksinkertaisella Java-ohjelmointikielellä toteutetulla sovelluksella. Sovellus käynnistää kolme säiettä (liite 1), joista jokainen simuloi yhtä laitetta ja lähettää kahden sekunnin välein tietoja palvelimelle MQTT-protokollaa käyttäen. MQTT-asiakasohjelmana käytetään avoimen lähdekoodin Eclipse Paho –kirjaston Java-versiota.

ThingsBoardin MQTT-rajapintaan julkaistaan viestit ”v1/devices/me/telemetry” -otsikolla. Itse viesti on JSON-muotoinen ja sisältää kolme attribuuttia arvoineen.

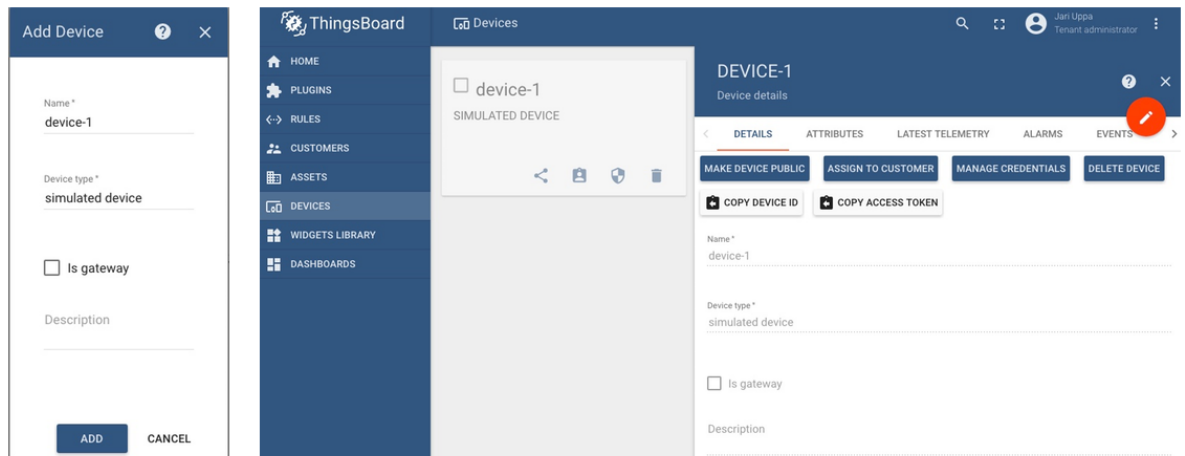
```
{'attr1' : 20, 'attr2' : 374, 'attr3' : 90}
```

Arvoiksi generoidaan satunnaiset kokonaisluvut käyttäen attribuuttikohtaisia vaihteluvälejä ilman sen tarkempaa logiikkaa, joten visualisointi valvontanäkymässä ei ole välttämättä mitenkään siistin näköinen. Ajatuksena on vain saada syötettyä jotain arvoja tutustumisen pohjaksi.

Jotta laitteet tunnistetaan ThingsBoard-järjestelmässä, täytyy MQTT-yhteys avata käyttäen käyttäjätunnuksena laitteelle hallintasovelluksessa rekisteröinnin yhteydessä generoitua ”access token” tietoa, joka on satunnainen merkkijono, esimerkiksi ”iLIEREmZT-SrpsHGNSi6N”.

6.2.2 Laitteiden rekisteröinti

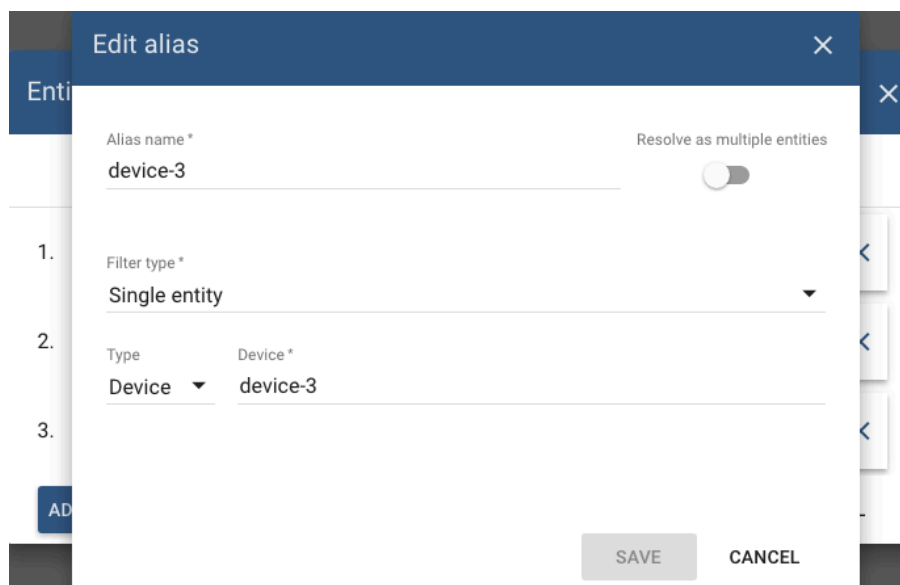
Järjestelmään tutustumisessa käytetään vain kolmea simuloitua laitetta, rekisteröidään ne hallintasovelluksella. Laitteelle annetaan nimi ja vapaasta valittavissa oleva tyyppi, jonka jälkeen laitteen tiedoista nähdään MQTT-yhteyden avaamiseen tarvittava tokeni. Laitteen tiedoista on myös nähtävissä sen lähettämän datan attribuutit ja niiden viimeisimmät arvot. Lisäksi näkymästä voidaan yhdistää laite tiettyyn asiakkaaseen ja hallinnoida siihen liittyviä hälytyksiä ja tapahtumia (kuva 13).



KUVA 13. Laitteen rekisteröinti hallintasovelluksessa

6.2.3 Valvontanäkymä

Jotta laitteiden lähettämä data saadaan näkymään valvontanäkymässä, täytyy ensin järjestelmään rekisteröidyille laitteille luoda alias-määrittelyt (kuva 14). Luodaan ensin valvontanäkymä ja sen jälkeen sille kolmea laitetta vastaavat laite-aliakset.



KUVA 14. Laite-aliaksen luonti

Tämän jälkeen valvontanäkymään voidaan lisätä vimpaimia, joille määritellään laite-aliasten avulla mistä laitteesta tai laitteista tuleva data näytetään (kuva 15). Lisäksi mää-

ritellään näytettävät attribuutit ja muuta ulkoasuun vaikuttavat määrittelyt, kuten esimerkiksi otsikot ja värit. Näkymän ollessa editointi-tilassa, voidaan siitä näytettävien vimpainten sijoittelua ja kokoa muuttaa hiirellä raahaamalla.

Näkymässä voidaan valita, kuinka tiedot näytetään ajan suhteen. Reaaliaikainen päivitys näyttää tiedot sitä mukaa kun niitä laitteilta saapuu. Historia-tilassa näytetään tiedot valitulta aikajaksolta, esimerkiksi viimeiseltä kolmen tunnin jaksolta tai joltakin kiinteältä aikaväliltä.

Tietoa voidaan myös käsitellä esitysvaiheessa sekä valvontanäkymän että joidenkin vimpaimen tasolla määrittelemällä haluttu aggregointi-tapa ja aikajakso. Attribuuttien arvoille voidaan laskea pienin ja suurin arvo, keskiarvo, lukumäärä tai summa valittua aikajaksoa kohti.

Add Widget

DATA SETTINGS ADVANCED ACTIONS

☒ Use dashboard timewindow

Timewindow **REALTIME - LAST MINUTE**

Datasources

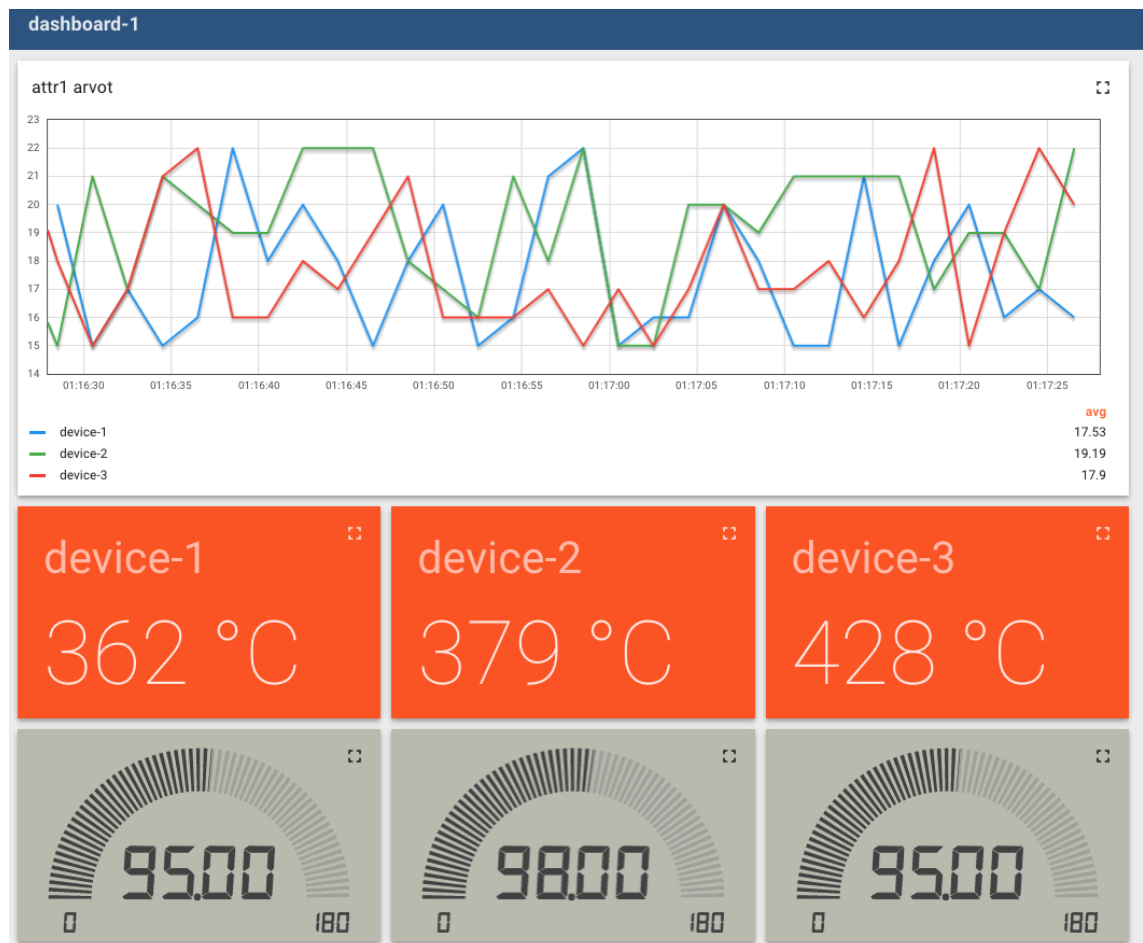
	Type	Parameters	
1.	Entity	device-1	attr1: attr1
2.	Entity	device-2	attr1 2: attr1
3.	Entity	device-3	attr1 3: attr1

ADD **CANCEL**

KUVA 15. Vimpain-määrittely

Jokaisella tuetulla vimpain-tyypillä on oma valikoimansa muokattavissa olevia asetuksia ja tietolähteitä, joiden avulla laitteiden lähettämään dataan päästään käsiksi. Tietoa näyttävien vimpainten lisäksi järjestelmä tukee kontrolli-vimpaimia, joiden avulla voidaan lähettää valvontanäkymästä RPC-ohjauskomentoja laitteille.

Kokeilun lopputuloksena syntyi valvontanäkymä, jossa näytetään kolmen järjestelmään liitetyn simuloidun laitteen lähettämien kolmen attribuutin arvot eri vimpaintyyppejä hyödyntäen (kuva 16). Tarkoitus oli lisätä näkymään vielä lisäksi vimpain, joka olisi näyttänyt hälytyksiä attribuuteille asetettujen rajojen ylityksistä, mutta tämän toiminnon konfigurointi ei onnistunut.



KUVA 16. Valvontanäkymä

6.2.4 Huomioita alustasta

ThingsBoard-alustaan tutustumisen yhteydessä kävi ilmi, että se on monilta osin vielä kehitystyön alla. Tuotteen websivuilta löytyvä dokumentaatio ei ollut kaikilta osin aivan

ajan tasalla joka vaikeutti järjestelmään tutustumista. Lisäksi jotkin toiminnot eivät toimineet hallintasovelluksessa odotetusti tai ei ollenkaan. Monessa tällaisessa tapauksessa esimerkiksi konfiguraatioiden määrittelyyn löytyi onneksi useampi tapa, jolloin puutteet eivät täysin estäneet etenemistä vaan ainoastaan hidastivat sitä. Järjestelmästä jäi kuitenkin kaiken kaikkiaan positiivinen mielikuva.

7 POHDINTA

Esineiden internetiin tutustumisen yhteydessä kävi hyvin selväksi, että opinnäytetyön puitteissa on mahdollista vain kevyesti raapaista pintaa ja yrittää selventää asiaan liittyviä peruskäsitteitä hyvin yleisellä tasolla. IoT-maailmaan liittyy niin monia erilaisia asioita, tuotteita ja tekniikoita, että aiheen käsittely vaati tarkkaa rajausta.

Opinnäytetyön tavoitteena ollut esineiden internetin peruskäsitteiden selventäminen ja IoT-järjestelmien toimintoihin ja markkinoilta löytyvään tarjontaan tutustuminen oli suhteellisen helppoa yleiskäsitteiden osalta, joista löytyy runsaasti vapaasti saatavilla olevaa materiaalia. Yhteystekniikoiden osalta tietoa on saatavilla kohtuullisen hyvin, mutta se on jo niin hyvää teknistä tietämystä vaativaa, että työssä niitä käsiteltiin hyvin pintapuolisesti. IoT-alustojen suhteen tietoa löytyy joistakin järjestelmistä hyvinkin runsaasti, jopa eräät kaupalliset tuotteet tarjoavat paljon teknistä tietoa joka on vapaasti saatavilla. Toisesta ääripäästä löytyvät tuotteet, joista tietoa ei ole juurikaan julkisesti tarjolla.

IoT-järjestelmiin tutustuminen oli erityisen mielenkiintoinen aihealue, suurin työ oli käydä läpi järjestelmiä ja etsiä sopiva tarkemmin esiteltäväksi. Ajatus oli kokeilla jotakin järjestelmän aivan käytännössä ja yrittää syöttää siihen jotakin dataa IoT-laitteelta ja selvittää mitä sille järjestelmässä voidaan tehdä. Alun perin oli tarkoitus käyttää Microsoft Azure IoT Hub -alustaa, mutta asiaa perehtymisen jälkeen totesin, että visualisointi-aspektin mukaan saaminen olisi vaatinut laajempaa tutustumista Azure-pilvipalvelun tarjoamiin tuotteisiin ja toimintoihin ja se olisi vaatinut hieman liikaa resursseja. Avoimen lähdekoodin tuotteiden joukosta erottui kuitenkin ThingsBoard-tuote, josta löytyi halutut ominaisuudet yhteen paketoituna ja se oli vielä kaiken lisäksi helppo asentaa ja ottaa käyttöön tutustumista varten. Tuote osoittautuikin todella mainioksi ja erittäin helppokäyttöiseksi, lisäksi se tarjoaa hyvän laajennettavuuden moniin erilaisiin käyttötarkoituksiin.

Suurin esineiden internetiin liittyvä huolenaihe liittyy tietoturvaan ja yksityisyyteen. Miljoonien laitteiden liittäminen verkkoon ei voi sujua ilman ongelmia ja tästä löytyi runsaasti erilaista materiaalia koskien jo ilmenneitä ongelmia. Monia laitteita ei ole suunniteltu varta vasten tietoturvallisiksi ja lisäksi käyttäjät omilla toimillaan aiheuttavat potentiaalisia tietoturvariskejä. IoT-laitteita voidaan käyttää erilaisiin verkkohyökkäyksiin ja sen lisäksi niiden kautta saattaa paljastua käyttäjien yksityisyyteen liittyvää informaatiota.

Esineiden internet vaikuttaa asiaan perehtymisen jälkeen vielä laajemmalla kokonaisuu-
delta kuin ennen työn aloittamista. Erilaisia mahdollisuuksia hyödyntää IoT-tekniikoita
on valtavasti ja tällä hetkellä on vaikea kuvitella mille kaikille elämän alueille IoT leviää
muutamien seuraavien vuosien kuluessa. Hyvin todennäköisesti sen erilaiset sovellukset
tulevat niin olennaiseksi osaksi jokapäiväistä elämäämme, että ne tuntuvat lopulta itses-
täänselvyyksiltä ja ihmettelemme, kuinka ilman niitä on aiemmin tultu toimeen.

LÄHTEET

Amazon IoT Platform, 2017. Luettu 3.9.2017. <https://aws.amazon.com/iot-platform>

Arm mbed, 2017. Luettu 1.10.2017. <https://www.mbed.com>

Aviation Week, 2016. Internet Of Aircraft Things: An Industry Set To Be Transformed. Luettu 14.9.2019. <http://aviationweek.com/connected-aerospace/internet-aircraft-things-industry-set-be-transformed>

BeagleBoard, 2017. Luettu 18.8.2017. <https://beagleboard.org>

Business Insider, 2016. Here are IoT trends that will change the way businesses, governments, and consumers interact with the world. Luettu 11.9.2017. <http://www.businessinsider.com/top-internet-of-things-trends-2016-1>

Business Insider, 2017. THE INTERNET OF THINGS 2017 REPORT: How the IoT is improving lives to transform the world. Luettu 11.9.2017. <http://www.businessinsider.com/the-internet-of-things-2017-report-2017-1>

Business Wire, 2015. Amazon Web Services Announces AWS IoT. Luettu 8.9.2017. <http://www.businesswire.com/news/home/20151008006261/en>

Boston Consulting Group, 2017. Winning in IoT: It's All About the Business Processes. Luettu 12.9.2017. <https://www.bcgperspectives.com/content/articles/hardware-software-energy-environment-winning-in-iot-all-about-winning-processes>

Google Cloud IoT, 2017. Luettu 25.9.2017. <https://cloud.google.com/iot>

Hanes D., Salgueiro G. & Barton R. 2017. IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things. Indianapolis: Cisco Press.

Intel, 2015. Could Smart Homes Be as Commonplace as Smartphones by 2025?. Luettu 10.9.2017. <http://download.intel.com/newsroom/kits/iot/pdfs/IntelSmartHomeSurveyBackgrounder.pdf>

IoT Analytics, 2017. IoT Platform Comparison: How the 450 providers stack up. Luettu 21.9.2017. <https://iot-analytics.com/iot-platform-comparison-how-providers-stack-up/>

Kaa IoT Platform, 2017. Luettu 5.7.2017. <https://www.kaaproject.org>

MarketsandMarkets, 2016. IoT Cloud Platform Market worth 7.15 Billion USD by 2021. Luettu 17.9.2017. <http://www.marketsandmarkets.com/PressReleases/iot-cloud-platform.asp>

MQTT. Luettu 21.9.2017. <http://mqtt.org/>

Libelium, 2016. Smart Airport project: monitoring environmental conditions in Santiago de Chile airport. Luettu 15.9.2017. <http://www.libelium.com/smart-airport-project-monitoring-environmental-conditions-in-santiago-de-chile-airport/>

Perry M. 2016. Evaluating and Choosing an IoT Platform. Sebastopol: O'Reilly Media, Inc.

The IoT World Forum, 2014. The Internet of Things Reference Model. Luettu 18.9.2017. http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf

ThingsBoard, 2017. Luettu 4.10.2017. <https://thingsboard.io/>

The Wall Street Journal, 2017. The Rise of the Smart City. Luettu 14.9.2017. <https://www.wsj.com/articles/the-rise-of-the-smart-city-1492395120>

Wired, 2017. We've reached peak IoT. There's now a smart hairbrush. Luettu 17.9.2017. <http://www.wired.co.uk/article/smart-hair-brush-loreal-withings>

LIITEET

Liite 1. Laitesimulaattori, MQTT-viestin lähetyt

```
package fi.uppa.thingsboard.devsim;

import org.eclipse.paho.client.mqttv3.MqttClient;
import org.eclipse.paho.client.mqttv3.MqttConnectOptions;
import org.eclipse.paho.client.mqttv3.MqttException;
import org.eclipse.paho.client.mqttv3.MqttMessage;
import org.eclipse.paho.client.mqttv3.persist.MemoryPersistence;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class Device implements Runnable {

    private Logger logger = LoggerFactory.getLogger(this.getClass().getName());

    private String MQTT_TOPIC = "v1/devices/me/telemetry";
    private String MQTT_BROKER = "tcp://<hostname or IP>:<port>";
    private int QOS = 1;

    private MqttClient mqttClient;

    public Device(String clientId, String username) {

        try {

            mqttClient = new MqttClient(MQTT_BROKER, clientId, new MemoryPersistence());

            MqttConnectOptions connOpts = new MqttConnectOptions();
            connOpts.setUserName(username);
            connOpts.setCleanSession(true);

            mqttClient.connect(connOpts);

        } catch (MqttException me) {

            logger.error(me.getMessage());

        }

    }

    public void run() {

        try {

            if (mqttClient != null && mqttClient.isConnected()) {

                int val1 = Utils.getRandomInteger(15, 22);
                int val2 = Utils.getRandomInteger(300, 500);
                int val3 = Utils.getRandomInteger(90, 100);

                String msg = String.format("'attr1' : %s, 'attr2' : %s, 'attr3' : %s",
                    val1, val2, val3);

                MqttMessage message = new MqttMessage(msg.getBytes());
                message.setQos(QOS);

                mqttClient.publish(MQTT_TOPIC, message);

            }

        } catch (MqttException me) {

            logger.error(me.getMessage());

        }

    }

}
```


Liite 2. Laitesimulaattori

```

package fi.uppa.thingsboard.devsim;

import java.util.Scanner;
import java.util.concurrent.Executors;
import java.util.concurrent.ScheduledExecutorService;
import java.util.concurrent.ScheduledFuture;
import java.util.concurrent.TimeUnit;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class Simulator {

    public static void main(String[] args) {

        Logger logger = LoggerFactory.getLogger("Simulator");

        ScheduledExecutorService scheduledExecutorService
            = Executors.newScheduledThreadPool(5);

        ScheduledFuture<?> future = scheduledExecutorService
            .scheduleAtFixedRate(new Device("dev1", "<access token>"),
                2, 2, TimeUnit.SECONDS);

        scheduledExecutorService
            .scheduleAtFixedRate(new Device("dev2", "<access token>"),
                4, 2, TimeUnit.SECONDS);

        scheduledExecutorService
            .scheduleAtFixedRate(new Device("dev3", "<access token>"),
                6, 2, TimeUnit.SECONDS);

        Scanner s = new Scanner(System.in);

        logger.info("Press enter to continue.....");

        s.nextLine();
        s.close();

        scheduledExecutorService.shutdownNow();
        logger.info("exit");
        System.exit(0);

    }

}

-----

package fi.uppa.thingsboard.devsim;

import java.util.Random;

public class Utils {

    public static int getRandomInteger(int min, int max) {

        Random r = new Random();
        return r.ints(min, (max + 1)).findFirst().getAsInt();

    }

}

```